

Optimal Scheduling for Interference Mitigation by Range Information

Vijaya Yajnanarayana*, Klas E. G. Magnusson, Rasmus Brandt, Satyam Dwivedi, Peter Händel

Abstract—This paper describes several algorithms for generating an optimal schedule for multiple access on a shared channel by utilizing range information in a fully connected network. We also provide detailed analysis for the proposed algorithms in terms of their complexity, convergence, and effect of non-idealities in the network. The performance of the proposed schemes are compared with non-aided methods to quantify the benefits of using the range information in the communication. We argue that the proposed techniques yield significant benefits as the number of nodes in the network increases. We provide simulation results in support of the claim. The proposed methods indicate that the throughput can be increased on average by 3 – 10 times for typical network configurations.

Index Terms—Sensor networks, ad-hoc networks, mobile networks, swarm networks, cooperative communication, position dependent communication, UWB communication, mmwave communication, 5G-communication.

I. INTRODUCTION

The recent advances in sensor technology have resulted in development of low-cost, low-power sensors, which are capable of sensing, data processing, and communication. Many sensor networks have a large number of sensor nodes, which are densely deployed over a wide geographical region to track a certain physical phenomenon [1], [2]. These sensors could have a fixed topology, as in the case of smart sensors used in structural health monitoring [3] or have a dynamic topology, as in the case of sensors mounted on autonomous robots for applications discussed in [4]–[6].

In sensor networks, there are many situations where every node needs to transmit information to every other node at regular intervals, through a shared channel. For example, in [7], [8] firefighter agents share information at regular intervals through point to multi-point communication, where every agent broadcasts sensor data, like position, temperature, visibility, etc., to all other agents. This enables every firefighter to know relevant information about other firefighters, thereby increasing the efficiency of operation. This is illustrated in Fig. 1. This type of communication can also be found in the cooperating swarm of micro unmanned aerial vehicles (UAVs). These are low payload carrying, scaled down quadrotor platforms with relevant sensors mounted on them [6]. Constant updates (communication) between sensors are essential in multiple UAVs network, as they need to coordinate to accomplish the required tasks. These updates could include sensor data, position information, etc. Fig. 2 shows a graphical depiction of quadcopters in a particular geometric formation. Similar regular broadcast communication by sensor nodes can also be

found in other swarm networks as discussed in [4], [5], [9]. Reporting the health of the sensor node to all other nodes in wireless sensor networks (WSN) as described in [10] also requires regular communication by the sensor nodes.

Sensor networks in which each sensor has to share information constantly with the other sensors in the networks can be accomplished efficiently by communicating through a shared channel. As the density of the sensor network increases, the effective bitrate per sensor, R_s , drops, since the total bitrate, R_b , supported by the shared common channel is fixed. Our intention in this paper is to develop an efficient communication method for the sensor network by exploiting the position information so that we can have higher throughput per sensor. We define one report cycle (update cycle), T_R , as the total time duration during which all the nodes in the sensor network have transmitted and received one message packet to and from all the other nodes in the network.

In a regular time division channel, the shared common channel is slotted in time and each one of the N nodes of the sensor network will have access to a time slot which is a uniform fraction of the report cycle, T_R . As shown in Section II, as the radius of the sensor network topology and the number of nodes in it increase, the throughput per sensor and the update rate decrease. By exploiting the range information, orthogonality can still be maintained for overlapping time slots which leads to higher capacity. For ideal positioning of nodes, the throughput can be increased by N times, leading to a significant performance gain. Even when the positions of nodes are randomly distributed, the performance boost can be substantial in practice. For the realistic examples studied in this paper, the throughput is increased by an order of magnitude compared with a regular scheme. The main purpose of this paper is to introduce, synthesize and analyze a communication scheme using range information in sensor networks. We propose a method to generate an optimal time schedule, where each node in the sensor network can access the common channel. Each node will transmit information packets in its time schedule using the physical layer communication signal.

We will demonstrate the methods using a simple 3 node network shown in Fig. 3. This will aid us in explaining the algorithms clearly. Subsequently, we will demonstrate the performance of the proposed methods in different network topologies of varied sizes. We use the report cycle, T_R , as a metric to assess performance, where the objective is to minimize this parameter. The rest of the paper is organized as follows: In Section II, we discuss the system model and formulate the problem. In Section III, we propose algorithms which exploit the range information to provide efficient

The authors are with the Department of Signal Processing, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden. E-mail: vpy@kth.se.

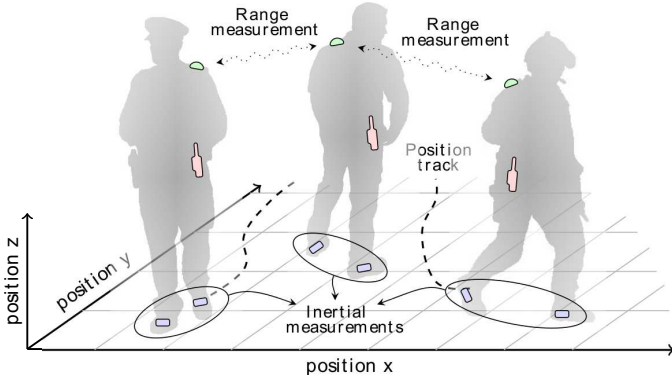


Fig. 1. Illustration of fire fighters agents sharing information continuously with other agents [7].

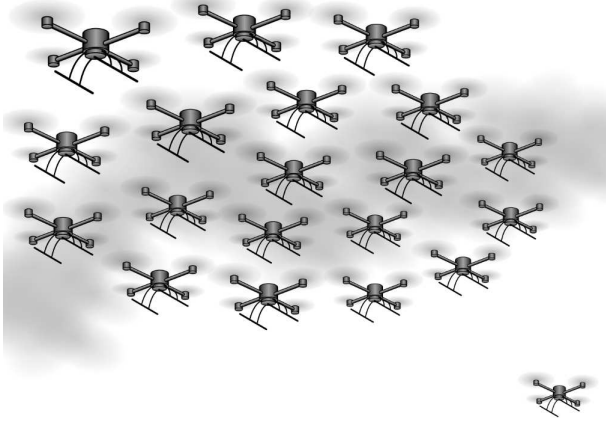


Fig. 2. A graphical illustration of a geometric configuration of a swarm network of micro quadcopters.

communication between nodes. In Section IV, we study the effect of synchronization and range errors on the proposed algorithms. In Section V, we compare the effective bitrate per sensor, R_s , of the proposed algorithms with the code division multiple access (CDMA) approach. In Section VI, we evaluate the proposed methods for a large number of nodes with different topologies and demonstrate the performance gain of utilizing the range information. Finally, in Section VII we discuss the conclusions.

II. SYSTEM MODEL AND PROBLEM FORMULATION

Consider a general setup of a fully connected sensor network with N nodes. For the sake of discussion, we set the access duration (message packet length) per node to be $\tau = 10$ time units. We define the path equivalent message length as $\mathcal{L} = \mu\tau$ length units, where μ is the velocity of the physical layer signal in the propagation medium. The message packets are said to be correctly received, if the packets do not interfere, i.e., there is no collision of packets at the receiving node.

While discussing the algorithms, we assume that all the sensor clocks are synchronized to the same master clock. In many sensor networks, synchronization is accomplished using a message passing technique as proposed in timing-sync (TSYNC) or reference broadcast synchronization (RBS)

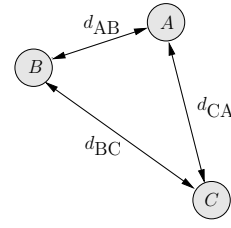


Fig. 3. Peer-to-peer ad-hoc sensor network with 3 nodes. d_{AB} , d_{BC} and d_{CA} are the path lengths between nodes A , B and C .

TABLE I
CONFIGURATION FOR THE TOPOLOGY IN FIG. 3.

Parameter	Value
d_{AB}	9.5 m
d_{BC}	11 m
d_{CA}	10.5 m
\mathcal{L}	3 m
τ	10 ns
μ	3×10^8 m/s

protocols [11], [12]. Network synchronization ensures that all the nodes in the network have the same time scale. We also assume that precise range information about the nodes is available. Recently, there has been some work on estimation algorithms for joint ranging and synchronization. These are proposed in [13], [14]. These algorithms can yield joint accuracy levels up to few centi-meters for range and few nano-seconds for synchronization. In practice, precise time synchronization and range estimation are not possible. We study the behavior of the proposed algorithms in the presence of range and synchronization errors in Section IV.

A. Orthogonalization with scheduled transmission.

In a network of N nodes, if we assume that the $K = \binom{N}{2}$ range values are available, one approach to orthogonalize the transmission is by creating a sequential schedule, where each node gets to transmit a message every T_D time units, where T_D is given by

$$T_D = \frac{D}{\mu} + \tau, \quad (1)$$

where D is the maximum of the K range values, that is

$$D = \max_{i,j} \{d_{ij}\}, \forall i, j \in [1, \dots, N], i \neq j. \quad (2)$$

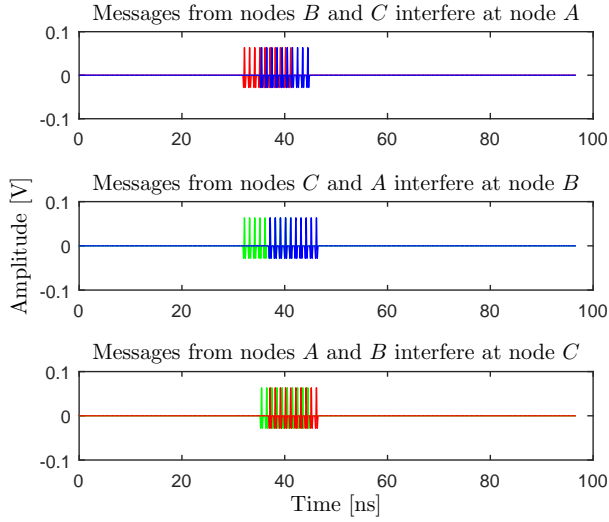
With this approach, one report cycle, T_R is given by

$$T_R = NT_D. \quad (3)$$

To exemplify the above discussion, we consider a 3 node peer-to-peer network as shown in Fig 3. For the sake of discussion, the nodes are labeled as A , B , and C . From (1) and (2), we get

$$T_D = \frac{\max\{d_{AB}, d_{BC}, d_{CA}\}}{\mu} + \tau = \frac{d_{BC}}{\mu} + \tau. \quad (4)$$

From (3), notice that the report cycle, T_R , increases linearly with the number of nodes in the network (N) and the radius of the network topology (D). Therefore, as the number of nodes or the geometric size of the network increases, T_R ,



(a) Interfering message packets due to concurrent transmission.

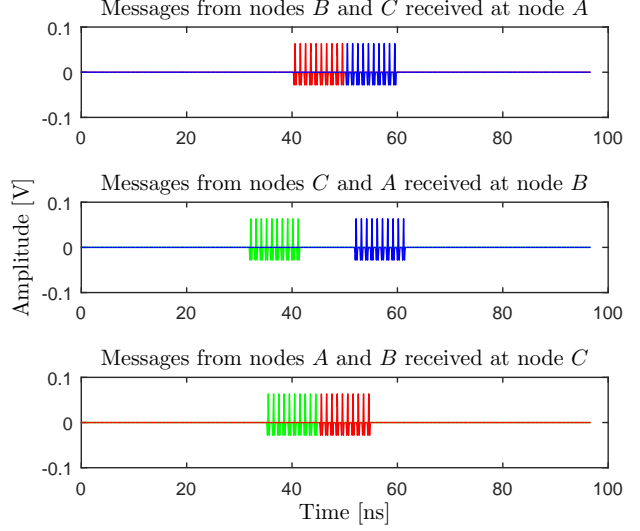
(b) Arrival of packets without interference at A , B and C nodes after introducing delays of $\Delta_B = 8.4$ ns and $\Delta_C = 15$ ns in B and C nodes, respectively.

Fig. 4. Concurrent transmission on shared common channel will result in interference as shown in (a). If we solve the optimization problem defined in (6) then the interference can be mitigated as shown in (b). The Message packet from nodes A , B and C is shown in green, red, and blue respectively. The $\tau = 10$ ns, $\mu = 3 \times 10^8$ m/s, and $\mathcal{L} = 3$ m is considered in the illustration.

will increase, resulting in inefficient utilization of the shared common channel; thus, requiring an improved communication method.

In many networks, the geometry of the sensor placements is such that the difference in propagation time for the message packets to arrive at nodes are larger than the duration of the message packets themselves. These situations arise in many sensor networks which have small message packets to be shared with other sensors, resulting in a very small value of \mathcal{L} . This situation could also arise in future 5G networks, where the physical layer packet lengths of devices in a macro cell are much smaller (on the order of a few microseconds) compared to the cell dimensions (on the order of a few kilometers) [15]–[18]. We can reduce the report cycle of the network by exploiting this fact. Consider a sensor network in which the path difference between any two nodes is greater than \mathcal{L} . Then, concurrent transmissions will result in message packets arriving at different times at each node, hence all transmissions are orthogonal. In general, for an N node network to ensure concurrent orthogonal transmissions, the network should fulfill the conditions

$$\begin{aligned} |d_{ki} - d_{kj}| &\geq \mathcal{L} \\ \forall i, j, k \in [1, 2, \dots, N] \mid i, j &\neq k \text{ and } i \neq j, \end{aligned} \quad (5)$$

where i, j and k denote the distinct nodes in the network and d_{ki} and d_{kj} denote the distance from the k -th node to node i and node j respectively. Thus, the report cycle, T_R , is equal to the maximum path delay, T_D , in the network, instead of NT_D for scheduled transmission as discussed before.

For example, consider the 3 node network shown in Fig. 3. Suppose, the dimensions of d_{AB} , d_{BC} , and d_{CA} does not follow the specifications of Table I and if $|d_{AB} - d_{AC}| \geq \mathcal{L}$, then the signal transmitted simultaneously at nodes B and C will

arrive at node A at different times, and hence A can correctly receive them. Similarly, $|d_{BA} - d_{BC}| \geq \mathcal{L}$ and $|d_{CA} - d_{CB}| \geq \mathcal{L}$ will ensure correct message packet reception at nodes B and C respectively. Thus, all the three nodes can concurrently transmit, and the report cycle can be completed in T_D .

In general sensor networks, (5) is rarely fulfilled. When a network with N nodes has a particular geometric configuration, which does not meet condition (5), we can reduce T_R by introducing a delay Δ_i to each node $i \in [1, 2, \dots, N]$. The Δ_i s are adjusted such that the message packets do not interfere at the receiving nodes. The Δ_i s form the time schedule during which the i -th node needs to transmit. The optimal schedule is obtained by solving the following optimization problem.

$$\begin{aligned} &\text{minimize } \max_{\{\Delta_i\}} \max_{i,k} \Delta_i + \delta_{ki} \\ &\text{subject to } J = J_1 + J_2 + \dots + J_N = 0, \end{aligned} \quad (6)$$

where

$$\begin{aligned} J_k &= \int \sum_{ij} |p_i(t - \delta_{ki} - \Delta_i)| |p_j(t - \delta_{kj} - \Delta_j)| dt \\ &\forall i, j, k \in [1, 2, \dots, N] \mid i, j \neq k \text{ and } i \neq j. \end{aligned}$$

Here, $p_x(t)$, $x \in [1, 2, \dots, N]$, denotes the physical layer signal of the message packet. The δ_{ki} represents the path-delay between the k -th node and i -th node and is given by d_{ki}/μ . The report cycle with this approach is given by

$$T_R = \max_{i,j} (\Delta_i + \delta_{ji}) + \tau, \forall i, j \in [1, \dots, N] \text{ and } i \neq j. \quad (7)$$

To illustrate the solution of the optimization problem (6), we once again consider the 3 node network shown in Fig. 3. The configuration defined in Table I is used for path lengths. In Table I, the path differences between nodes do not meet the

constraint defined in (5). That is, if all the nodes transmit simultaneously, they will interfere with each other. For example, if at time $t = 0$, all the nodes A , B and C concurrently transmit their message packets, then the received signal at nodes A , B and C are shown in Fig. 4a.

To accomplish short report cycle without interference in the example discussed above, the optimization (6) is solved using the grid search method with $\tau = 10$ ns, $\mu = 3 \times 10^8$ m/s, and $\mathcal{L} = 3$ m. In this method, we set $\Delta_A = 0$; assuming that all nodes are synchronized to node A , J is computed by varying Δ_B and Δ_C over the interval $[0, T_D]$, where T_D is given by (1). The solution for the optimization problem using the grid search method yields $\Delta_B = 8.4$ ns and $\Delta_C = 15$ ns. With these delays introduced in nodes B and C , the signals are not interfering, as shown in Fig. 4b. Node C , will transmit last after a delay of 15 ns and the resulting report cycle using (4) is 47 ns.

III. ALGORITHMS

Using the grid search method to solve (6) is costly, as the algorithm complexity, $O(q^N)$, increases exponentially with the number of nodes in the network. Here, q indicates the size of the quantized grid of interval $[0, T_D]$ used in the grid search. In this section, we propose three distinct methods to solve the above problem.

A. Convex Algorithm (CA)

The optimization problem defined in (6) is not a convex problem since the equality constraints are not affine. The problem can however be made convex by introducing additional constraints.

Consider the arrival of messages at node k from nodes i and j as shown in Fig. 5. We can treat the arrived message packets as boxes of width τ , and thus the message packets will not interfere if the corresponding boxes do not overlap. If we have predetermined the order in which the message packets should arrive at a particular node, we can make sure that the corresponding boxes do not overlap using a simple linear inequality. For example, in Fig. 5, the inequality would be $\Delta_i + \delta_{ki} + \tau \leq \Delta_j + \delta_{kj}$. Thus, we have isolated the non-convexity of the optimization problem into selecting the order in which the message packets should arrive at the different nodes. Suppose, we consider a sequential schedule, in which node $i + 1$, will transmit after node i , then we can construct the optimization problem as

$$\text{minimize } \max_{\{\Delta_i\}} \Delta_i + \delta_{ki} \quad (8)$$

$$\text{subject to} \quad \Delta_i + \delta_{ki} + \tau \leq \Delta_{i+1} + \delta_{k,i+1} \quad (9)$$

$$\Delta_i \geq 0 \quad (10)$$

In (9), i goes from 1 to $N - 1$, as there is no node with index $N + 1$, and $k \neq i, i + 1$.

This is a convex optimization problem, as the objective function is convex, and all the inequality constraints are convex. The problem can be solved as a general linear program [19], [20], but algorithms with lower complexity can be

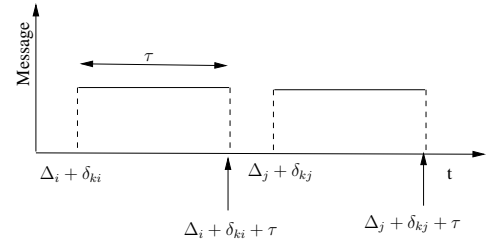


Fig. 5. Messages from node i and j arriving at node k

constructed by exploiting the structure of the problem. We found that the most efficient way to solve (8) is to minimize the delays Δ_i sequentially in order of increasing i . We note that Δ_{i+1} is minimized when it is zero or when (9) is tight for at least one k . For the first node, the smallest possible delay is $\Delta_1 = 0$ and for subsequent nodes the smallest possible delays are given by

$$\Delta_{i+1} = \max \left\{ 0, \Delta_i + \max_k \{ \delta_{ki} - \delta_{k,i+1} \} + \tau \right\}. \quad (11)$$

This results in a solution where none of the delays can be decreased without violating either (9) or (10), meaning that we have found an optimum of (8). The algorithm can be thought of as sliding the boxes corresponding to transmission $i + 1$ to the left along the time axis until one of them hits 0 or a box from transmission i .

In the above formulation, we have only considered interference between messages from nodes which come directly after each other in the node order. Given that node i does not receive a message from itself, it may be possible for messages from node $i - 1$ and node $i + 1$ to interfere when they are received at node i . This can however never happen, as (9) implies that

$$\Delta_{i-1} + \delta_{i,i-1} + \tau \leq \Delta_{i+1} + \delta_{i,i+1} \quad (12)$$

for $i = 2, 3, \dots, N - 1$. This is shown in Appendix A. Given that N delays need to be computed and that N path delays must be considered in each computation, the algorithm has a complexity of $O(N^2)$.

If the node order is set to A, B, C, in the configuration defined in Table I, this method produces the same solution as the grid search method, within the grid search tolerance¹. Even though the formulated problem is convex, for the N -node scenario, sequential ordering may not be the optimal order with the lowest report cycle. Selecting an optimal order is in itself a combinatorial optimization problem [21], [22]. However, for most practical scenarios, we can select an arbitrary order and solve the convex problem as demonstrated in Section VI.

B. Optimizing the node order by solving a TSP

The problem of selecting a good node order can be formulated as an asymmetric traveling salesman problem (TSP) [23], where the cost matrix is derived from the path delays. To

¹Note that the convex solution does not always produce the optimal solution, and thus may not always match the result from the grid search method.

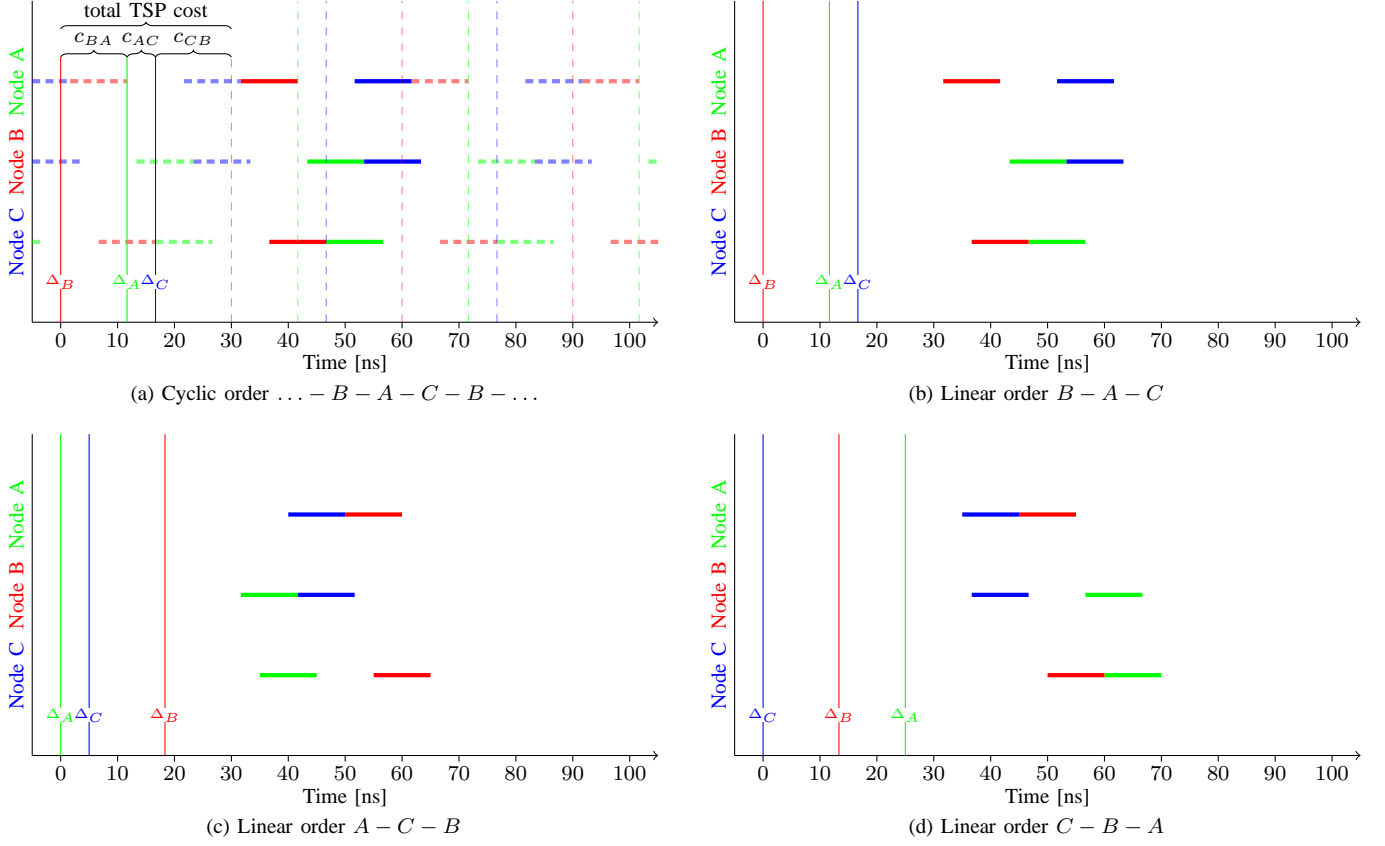


Fig. 6. The cyclic TSP solution (a) and the 3 possible linear orders that can be created from it (b)–(d), for the network in Fig. 3. The filled in horizontal bars show one cycle of received messages. The times of transmission are shown as solid vertical lines. In previous and future cycles, received messages and times of transmission are shown as dashed bars and dashed lines respectively. The TSP-costs along the cheapest tour can be visualised as the time differences between the transmissions. The duration of one cycle is 30 ns. In (b), (c), and (d), the report cycles are approximately 63.3 ns, 65.0 ns, and 70.0 ns respectively.

be able to do this, we modify the problem so that the nodes transmit in a cyclic order where a second message from the first node is placed directly after the first message from the last node. Then we solve a TSP problem which minimizes the time between two transmissions from the same node. Finally, we consider the N different ways in which the cyclic order can be broken into a linear order, and select the alternative which minimizes the report cycle in the original problem.

The objective of the traveling salesman problem is to find the cheapest tour which visits a number of cities exactly once. The input to the problem is a cost matrix, \mathbf{C} , where its element c_{ij} is the cost of going from city i to city j [23]. In our problem, we let each city correspond to a node in the network. We define the cost matrix so that c_{ij} is the minimum difference between the delays of node j and node i , allowed by (9), given that j comes directly after i in the node order. Given that we are looking at a cyclic order, node 1 takes the role of node $N + 1$ in (9), and we do not need to take the constraints (10) into consideration. If node j comes directly after node i in the selected order, we have that

$$\Delta_j = \Delta_i + \max_k \{\delta_{ki} - \delta_{kj}\} + \tau. \quad (13)$$

In other words, the delay of any node is equal to the delay of the previous node, plus the cost

$$c_{ij} = \max_k \{\delta_{ki} - \delta_{kj}\} + \tau. \quad (14)$$

By adding up all of the costs associated with the successive node pairs in the transmission order (TSP tour), we therefore get the time between two transmissions made by the same node. The problem of minimizing the time between two transmissions made by the same node can therefore be formulated as a TSP where the cost matrix is defined by c_{ij} . For the 3 node configuration shown in Fig. 3, the algorithm is graphically illustrated in Fig. 6.

The TSP is known to be NP-hard, but there are algorithms that can find exact solutions for small problems, and other algorithms that can find approximate solutions for larger problems [24], [25]. Many techniques employ heuristic approaches for finding the approximate solution [26]–[28]. The best approximate algorithms, often produce optimal or very close to optimal solutions, for large networks with hundreds of nodes. Furthermore, the approximate algorithms can be run multiple times with different starting points and thereby achieve much better performance [29]. We have chosen to use the TSP solver LKH [26], which is based on the Lin-Kernighan heuristic. For a problem with 100 nodes, LKH requires less than a second to produce a solution which has a high probability of being optimal. In LKH, all of the costs in matrix \mathbf{C} , must be integers and therefore we mapped the costs in each problem to the the interval between 0 and 10^6 using an affine mapping and rounded them to the closest integers.

We used version 2.0.7 of LKH with the default settings for all problems.

The report cycle will depend on which node in the TSP cycle is selected as node 1. Therefore we consider all of the N possible choices for node 1 and solve the convex problem defined in Section III-A for each one of them to see which alternative results in the shortest report cycle. Given that we can reuse the costs that we computed in (14) when we compute the delays in (11), the problem of choosing a first node has complexity $O(N^2)$. The overall complexity is therefore dominated by the TSP solver, which has an average complexity that scales approximately as $O(N^{2.2})$ [26]. We may introduce some sub-optimality by transforming the problem into a problem with transmitters in a cyclic order, and LKH may also not find the exact optimum of the TSP. The gap to optimality would however be negligible for most practical applications.

C. Iterative path-adjusting algorithm (IPA)

The CA reduces the algorithm complexity by allowing sub-optimality due to the fixed ordering. On the other hand, the TSP algorithm improves over the CA, by choosing a better order without increasing the algorithmic complexity significantly. One problem with both the algorithms is their inefficiency when (5) holds for most of the nodes (i.e., nodes are scattered far-apart compared to \mathcal{L}). For a random node configuration, we can in theory ensure that (5) holds by making the message length τ small enough. If τ is decreased by $d\tau$, the report cycle of the algorithms will however only decrease by $Nd\tau$, as the algorithms cannot change the order in which messages are to be received at the nodes. This results in poor performance when τ is small in comparison to the path delays of the network. To overcome this problem, we propose an alternative algorithm called iterative path-adjusting algorithm (IPA). We show in the later sections that this algorithm outperforms the convex formulation with strict ordering as defined in (9), and the TSP algorithm, when the sensor nodes are scattered wide apart.

In this algorithm, we adjust the path differences between nodes, d_{ki} and d_{kj} to satisfy (5) in an iterative way. Adjusting the path difference is the same as introducing delays at nodes i and j , so that the signals from i and j do not interfere at node k . The algorithm is described below in three steps followed by an example on a 3-node network.

- 1 Start the first iteration with $l = 0$ ($l + 1$ denotes the iteration number) and $k = 1$, with $d_{ki}^0 = d_{ki}$. For a topology having N nodes, add additional path lengths $d_{\Delta ik}^{l+1}$ and $d_{\Delta jk}^{l+1}$, $\forall i, j \in [1, 2, \dots, N]$, $i, j \neq k$, $i \neq j$ to nodes i and j to satisfy (5). Thus, the new path lengths are given by

$$d_{ki}^{l+1} = d_{ki}^l + d_{\Delta ik}^{l+1}, \quad (15)$$

$$d_{kj}^{l+1} = d_{kj}^l + d_{\Delta jk}^{l+1}. \quad (16)$$

Note that to satisfy (5), the path-length needs to be added to one of the nodes i or j . In this algorithm, we set $d_{\Delta ik}^{l+1} = 0$ and add additional path length $d_{\Delta jk}^{l+1}$ only to node j .

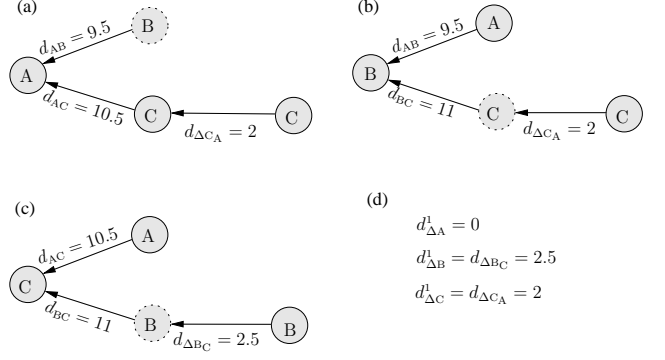


Fig. 7. Iteration 1 for the 3 node network shown in Fig. 3 with the configuration defined in Table I.

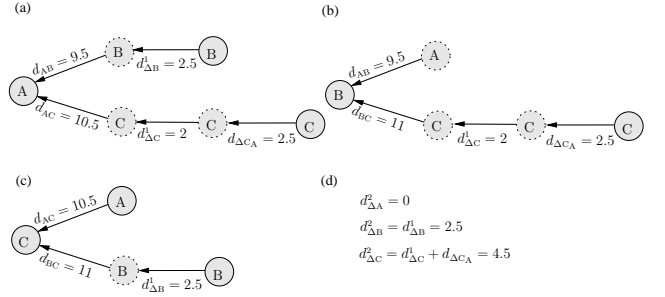


Fig. 8. Iteration 2 for the 3 node network shown in Fig. 3 with the configuration defined in Table I.

- 2 Repeat Step 1, by selecting all nodes one by one ($k = 1, 2, \dots, N$) in the network. Each time, carry over additional path lengths added $d_{ki}^l + d_{\Delta ik}^{l+1}$ and $d_{kj}^l + d_{\Delta jk}^{l+1}$. The total adjusted path lengths at the end of iteration l are given by

$$d_{\Delta k}^{l+1} = d_{\Delta k}^l + \sum_i d_{\Delta ik}^{l+1}, \quad (17)$$

for $k \in [1, 2, \dots, N]$ and $k \neq i$. This completes an iteration.

- 3 Repeat Step 1 and Step 2 until the total adjusted path length for each node does not change across iterations, meaning that the following condition holds for all $k \in [1, 2, \dots, N]$.

$$d_{\Delta k}^{l+1} = d_{\Delta k}^l. \quad (18)$$

This indicates that (5) is met for all nodes simultaneously.

The proposed method is illustrated with the 3 node network shown in Fig. 3, with the configuration defined in Table I. Fig. 7 (a) shows that the addition of an additional path length of $2 = (3 - (d_{AC} - d_{AB}))$ is required at node C to meet the constraints in (5) so that signals from B and C do not collide at A . Similarly, Fig. 7 (b) and Fig. 7 (c) add additional path lengths to the previous topology to avoid collisions at nodes B and C respectively. At the end of the 1st iteration, the total path lengths for all of the nodes are given in Fig. 7 (d).

The second iteration is illustrated in Fig. 8. Notice that we carried the new topology with added path lengths from the previous iteration ($d_{\Delta A}^1, d_{\Delta B}^1, d_{\Delta C}^1$) to iteration 2 and

at the end of iteration 2, the total added path lengths are $(d_{\Delta A}^2, d_{\Delta B}^2, d_{\Delta C}^2) = (0, 2.5, 4.5)$. Now the iteration is stopped as it meets the conditions defined in Step 3.

Translating the path lengths into path delays by dividing by the speed of light, c , results in $(\Delta_A, \Delta_B, \Delta_C) \approx (0, 8.4, 15)$ [ns]. This is the same result as with the grid search and convex methods. This algorithm is analyzed in the next Section.

D. Analysis of IPA

In order for the arriving signals not to interfere, (5) needs to be satisfied. We define the path matrix, \mathbf{M} , where each element of \mathbf{M} , d_{ki} , denotes the distance between node k and node i . The IPA adjusts the path matrix in such a way that the path lengths to node k from other nodes (represented by the k -th row in the matrix \mathbf{M}), have path differences greater than \mathcal{L} . For an N node network, the algorithm starts with the original path matrix, \mathbf{M}^0 , as given in (19). The path adjusted matrix after the l -th iteration is represented as \mathbf{M}^l .

$$\mathbf{M}^0 = \begin{bmatrix} 0 & d_{12} & \cdots & d_{1N} \\ d_{21} & 0 & \cdots & d_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ d_{N1} & d_{N2} & \cdots & 0 \end{bmatrix}. \quad (19)$$

Note that $d_{kk} = 0, \forall k \in (1, 2, \dots, N)$. Also, matrix \mathbf{M}^0 is symmetric, that is $d_{ki} = d_{ik}$. The path adjusted matrix, \mathbf{M}^l , has elements, d_{ij}^l . The i -th row of the path adjusted matrix \mathbf{M}^l is denoted as

$$\underline{d}_{ix}^l = [d_{i1}^l \quad d_{i2}^l \quad \cdots \quad d_{iN}^l], \quad (20)$$

similarly, the i -th column is denoted as

$$\underline{d}_{xi}^l = [d_{1x}^l \quad d_{2x}^l \quad \cdots \quad d_{Nx}^l]^T, \quad (21)$$

where τ denotes the transpose operator.

To perform step 1 of the algorithm, there are many possibilities for additional path lengths $d_{\Delta i_k}^l$ and $d_{\Delta j_k}^l$, such that the arriving signals at node k , have effective path length difference greater than \mathcal{L} . In this paper, in order to make the arriving signals to node k from nodes i and j satisfy $|d_{ki}^l - d_{kj}^l| \geq \mathcal{L}$, we will add path lengths only to j , if $j > i$. That is,

$$\text{if } |d_{ki}^l - d_{kj}^l| < \mathcal{L} \text{ and } j > i \text{ then} \quad (22)$$

$$d_{\Delta i_k}^{l+1} = 0, \quad (23)$$

$$d_{\Delta j_k}^{l+1} = \mathcal{L} - (d_{ki}^l - d_{kj}^l), \quad (24)$$

$$d_{xj}^{l+1} = [d_{xj}^l + d_{\Delta j_k}^{l+1} \mathbf{1}], \quad (25)$$

$$\forall i, j \in [1, 2, \dots, N], \quad i, j \neq k, i \neq j, \text{ and } j > i.$$

Where, $\mathbf{1}$ is $[1, 1, \dots, 1]^T$ and the process, defined in (22) to (25) is repeated for $k = 1, 2, \dots, N$ sequentially to complete an iteration. The iterations with $l = 0, 1, 2, \dots$ are performed until in (24), $d_{\Delta j_k}^{l+1} = 0, \forall i, j, k \in [1, 2, \dots, N], i, j \neq k$ and $i \neq j$ is met. At each iteration, the elements from \mathbf{M}^l are used for (22) to (25).

During each iteration, when (22) is met, the additional path is added only to one of the nodes (the node on the right).

TABLE II
SUMMARY OF AVERAGE COMPLEXITY OF THE PROPOSED METHODS.

Algorithm	Average case complexity
Grid Search	$O(q^N)$ (q is the size of the quantized grid)
CA	$O(N^2)$ (After exploiting the structure in the LP problem)
TSP	$O(N^{2.2})$ (Using LKH solver)
IPA	$O(N^3)$

Therefore, as the iterations increase, the path adjusted matrix, \mathbf{M} , will converge to the state with its elements

$$|d_{ki}^{l*} - d_{kj}^{l*}| \geq \mathcal{L}, \quad (26)$$

$$\forall i, j, k \in [1, 2, \dots, N], i, j \neq k \text{ and } i \neq j,$$

where, $l^* + 1$, denotes the number of iterations required for convergence. At this state the arriving signals to any node k from nodes i and j will satisfy (5). The effective adjusted path is given by

$$d_{\Delta i}^{l*} = \underline{d}_{1x}^{l*}(i) - \underline{d}_{1x}^0(i), \quad (27)$$

and the equivalent added delay for node, i , is $\Delta_i = d_{\Delta i}^{l*} / \mu$.

The average algorithmic complexity for IPA is evaluated by least square polynomial fit to the average computational time (in ticks) consumed by the algorithm for networks of different sizes. The procedure followed, along with the results, are discussed in Appendix B. From the results of the Appendix B, the average complexity of the IPA algorithm is $O(N^3)$.

A summary of the average complexities of the proposed methods is shown in Table II. The IPA opens up for a higher flexibility regarding the order of the transmissions. Thus, it provides better throughput compared to the convex algorithm as shown in Section VI. In mobile sensor networks, the convex approach with fixed ordering among the nodes opens up for schedule-based communication and ranging; thus, node information need not be encoded in the packets [30]. On the other hand, IPA requires transmission overhead since the node information has to be included in the packets, as the order of packet reception is not predetermined. However, the overhead of encoding the node information in the packet ($\log_2 N$ bits) is not significant.

The performances of the orthogonalization, CA, TSP, and IPA under large scale networks with different geometric formations are studied in the Section VI.

IV. EFFECT OF SYNCHRONIZATION AND RANGE ERRORS

In the discussion so far, we assumed that the sensor clocks are synchronized and the available set of range estimates (d_{ij} s) are accurate. Accurate network synchronization can be achieved by synchronizing the sensor clocks to the global positioning system (GPS) clocks or in GPS deficient systems by using the protocols discussed in Section II. Accurate range estimation can be accomplished by using TOA methods. The best performance in terms of mean-square-error (MSE) for an unbiased estimator is given by the Cramer Rao lower bound (CRLB) and for a time of arrival (TOA) estimation problem

this is given by [31], [32]:

$$\sigma_\tau^2 \geq \frac{1}{8\pi^2 \text{SNR} \beta^2}, \quad (28)$$

where β , is the effective signal bandwidth defined by

$$\beta^2 = \left[\frac{\int_{-\infty}^{\infty} f^2 |S(f)|^2 df}{\int_{-\infty}^{\infty} |S(f)|^2 df} \right], \quad (29)$$

where $S(f)$, is the Fourier transform of the transmit pulse, $s(t)$. Since many technologies like ultra wideband (UWB) use extremely large bandwidths, they can be used for precise range estimation. Practical UWB hardware with ranging and communication capabilities with range estimation accuracy of a few centimeters are discussed in [33], [34].

In practice, clock synchronization is not perfect and there will be range errors. These will result in message packets colliding at the receiving nodes. The synchronization error can be approximated to a zero mean normal distribution as shown in [35], [36]. In wideband RF systems [37], [38], problems such as multi-path fading, background interference, and irregular signal propagation characteristics make range estimates inaccurate. The range error can also be approximated to a zero mean normal distribution [39], [40].

We assume that synchronization and range errors are independent and the net effect will result in the packet arrival time to be randomly shifted from the intended position. The distribution of this random shift from the true position can be approximated to a Gaussian distribution, $\mathcal{N}(0, \sigma_e^2)$. This shift in time of arrival of the message packets at the receiving node can cause interference due to packet collisions. This problem can be reduced by adding a guard interval, ϵ , to the equations of the CA, TSP, and IPA. This can be done by expanding the message packet length τ to $\tau' = \tau + \epsilon$ in (9) and (5). Where ϵ can be used to trade off between tolerable interference and the report cycle time (update rate). We can show that

$$\epsilon = \sqrt{2}\sigma_e \text{erfc}^{-1}(2(1 - \mathcal{P})), \quad (30)$$

where $1 - \mathcal{P}$ denotes the percentage during which neighboring packets collide due to the range and synchronization errors. For example, to have 95% collision avoidance between neighboring packets, we need to have $\epsilon = 1.65\sigma_e$. The network level performance in presence of range and synchronization errors, using the above method, for proposed algorithms are studied in Section VI.

V. COMPARISON WITH CDMA SYSTEMS

Code division multiple access (CDMA) can be used for the type of sensor communication discussed in this paper. In this scheme, each node will be assigned a unique orthogonal code, \mathbf{u} , which is used to spread the message packet. Since the codes are orthogonal, nodes can concurrently transmit the message packets of width τ and thus the report cycle will be equal to the maximum propagation time T_D .

For an N -node sensor network, the length of the code, M , should be greater than N to ensure that the received packets are orthogonal at all nodes. Therefore, if we assume that the

shared common channel can support a bitrate of R_b [bps], then the effective bitrate per sensor, R_s , is given by

$$R_s^{\text{CDMA}} = \frac{R_b \tau}{MT_D}. \quad (31)$$

For the proposed algorithms in the paper, in each report cycle, T_R , each node in the network will get to transmit a message packet once for the duration, τ , seconds. Therefore the effective throughput per sensor can be computed as

$$R_s = \frac{R_b \tau}{T_R} \quad (32)$$

In the Section VI, we demonstrate in simulation the effective bitrate per sensor, R_s , as a function of the number of nodes, N , to show how the position information exploited in the proposed algorithms offer better performance compared to a CDMA based approach.

VI. SIMULATION STUDY

In the beginning of Section II-A, we mentioned that we can orthogonalize the message packets by separating consecutive transmissions by a time interval equal to the maximum path delay in the network. For example configuration in Table I, the report cycle, T_R , can be computed as below.

$$T_D = \frac{\max(d_{AB}, d_{BC}, d_{CA})}{\mu} + \tau = 47 \text{ ns}. \quad (33)$$

$$T_R = N \cdot T_D = 141 \text{ ns}. \quad (34)$$

However, if the path difference between nodes in the network topology satisfies (5), then all the nodes can concurrently transmit; thus one report cycle can be completed in the time duration equal to the maximum path delay in the network plus the packet length, that is 47 ns. More often (5) is not met. Under these circumstances we can minimize the report cycle by solving (6). We modified the problem so that it can be casted as a convex optimization problem. For the configuration in Table I, we showed that, ($\Delta_A = 0$ ns, $\Delta_B = 8.4$ ns, $\Delta_C = 15$ ns) solves (8), therefore node C will transmit last after a delay of 15 ns and complete the report cycle. So one report cycle for the configuration in Table I is

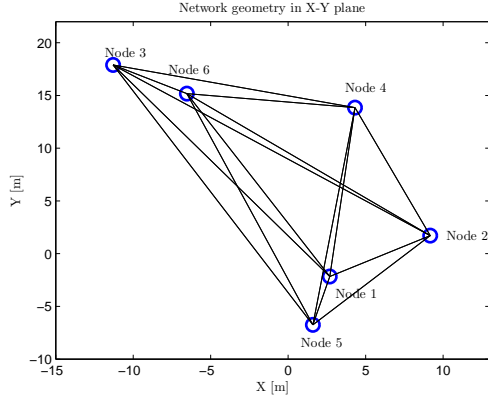
$$T_R = \max_{ij} (\Delta_i + \delta_{ji}) + \tau, \quad (35)$$

$$\forall i, j \in [A, B, C] \text{ and } i \neq j$$

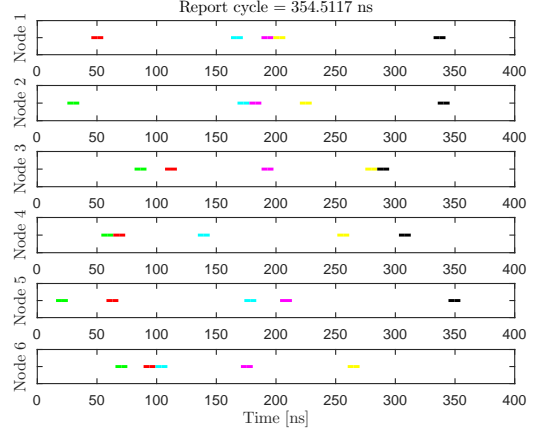
$$T_R = 15 + 37 + 10 = 62 \text{ ns} \quad (36)$$

Thus, the reduction in the report cycle equals 56%.

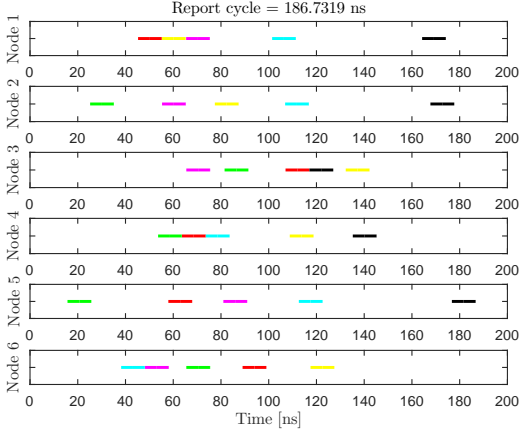
To study the performance of the proposed methods for large scale analysis, we form two different formations; one with outliers, where a few sensor nodes are far apart from the rest; and another with no-outliers, where the sensor nodes are scattered uniformly. The performance is reported in terms of the time required to complete one report cycle using the proposed methods.



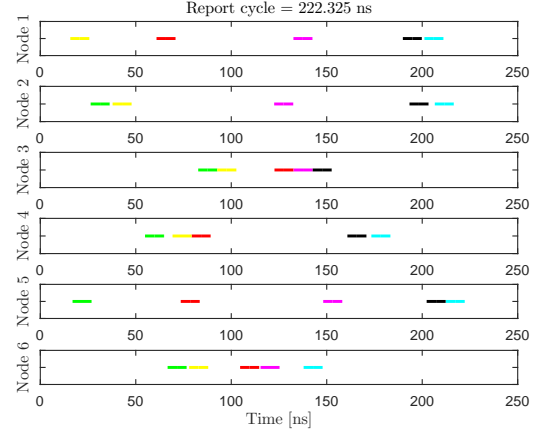
(a) Network Topology



(b) Convex Algorithm



(c) IPA



(d) TSP

Fig. 9. Network geometry with 6 nodes scattered randomly in a 2-D plane. Received message packets at each node after introducing the computed delays from Table III for network topology in Fig. 9a are shown for different algorithms. Notice that the message packets do not interfere. The color of the message packet is mapped to the node as shown in Table IV

A. Random geometric formation with no outliers

For performance analysis with no outliers, we create a random geometric formation by scattering the nodes in a plane. The coordinates (x, y) are drawn from a Gaussian distribution as shown below.

$$(x, y) \sim (\mathcal{N}(0, \sigma^2), \mathcal{N}(0, \sigma^2)) \quad (37)$$

A typical topology of 6 nodes with $\sigma = 5$ [m] is shown in Fig. 9a. The transmission schedules for interference mitigation, using different proposed algorithms are given in Table III.

With these delays introduced, the received packets will not interfere with each other. The received packets at each node are shown in Fig. 9b, 9c, and 9d for CA, IPA and TSP algorithms proposed in this paper. Each color in Fig. 9 is mapped to the messages from a specific node, as shown in the Table IV.

Report cycle, T_R , for the given set of delay values computed using CA, TSP and IPA are given by $\max_{i,j}(\Delta_i + \delta_{ij}) + \tau$. For the example network shown in Fig. 9a, the report cycles are given by 354.51 ns, 222.32 ns, and 186.73 ns for CA, TSP, and IPA respectively.

TABLE III
COMPUTED DELAY VALUES FROM PROPOSED METHODS FOR THE GEOMETRIC FORMATION DEFINED IN FIG. 9A.

Node	Delay Values (Δ_i s) [ns]		
	CA	IPA	TSP
1	0	0	1.19
2	20.09	20.08	35.68
3	80.78	19.92	119.63
4	134.17	11.74	78.90
5	182.13	39.57	0
6	266.56	98.78	124.38

From Fig. 9b and Fig. 9c, notice that the IPA is less constrained than the convex approach; the convex formulation requires that the order of the received message packets is the same at each receiving node. This is not the case for the IPA algorithm. This ensures tighter schedules and explains the better performance of the IPA algorithm.

To assess the performance over a large number of nodes N , we performed Monte-Carlo simulations. We swept the number of nodes, N , from 10 to 100 in steps of 10 and for each N , 32 distinct random geometric formations were constructed as

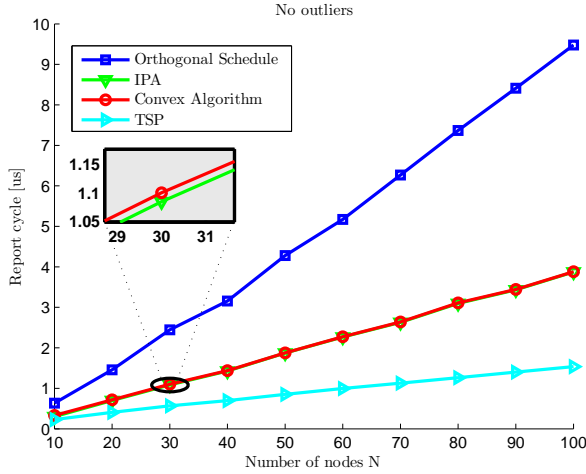


Fig. 10. Performance of proposed methods as a function of N . Notice that as the number of nodes increases, the proposed techniques yield better performance relative to the orthogonal schedule.

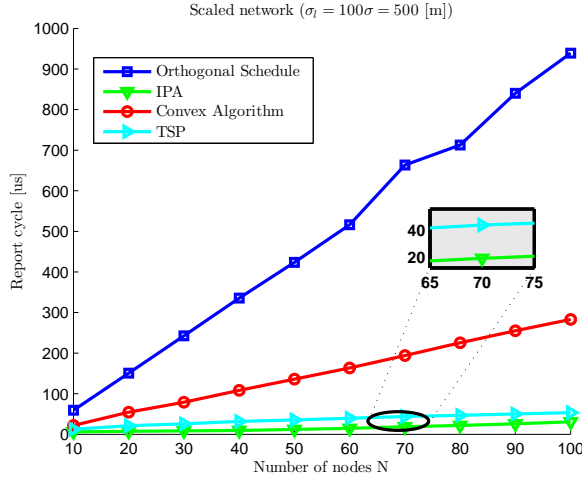


Fig. 11. Performance of proposed methods as a function of N . Notice that as the number of nodes increases, the IPA algorithm yield better performance relative to TSP when the network radius is larger compared to packet length.

per (37). The averaged report cycle is reported in Fig. 10.

Fig. 10 compares the proposed algorithms to the technique of orthogonalization with scheduled transmission discussed in Section II-A. Notice that for a 100 node randomly scattered network with $\sigma = 5$ [m], T_R is reduced to approximately 1/10 for the TSP algorithm and 1/3 for the fixed order convex algorithm and the IPA algorithm. However, if the radius of the network is scaled by a factor of 100 by changing the variance $\sigma_l = 100\sigma$, the IPA performs better than the TSP algorithm as explained in the earlier section and confirmed in simulation by Fig. 11.

The effective rate per sensor for the CDMA approach and the proposed algorithms are as given by (31) and (32). Fig. 12, shows the rate per sensor for CDMA and the proposed algorithms, assuming $R_b = 1$ Gbps and $\tau = 10$ ns. Notice that the proposed algorithms yield better performance in terms of bitrate per sensor, R_s , compared to CDMA.

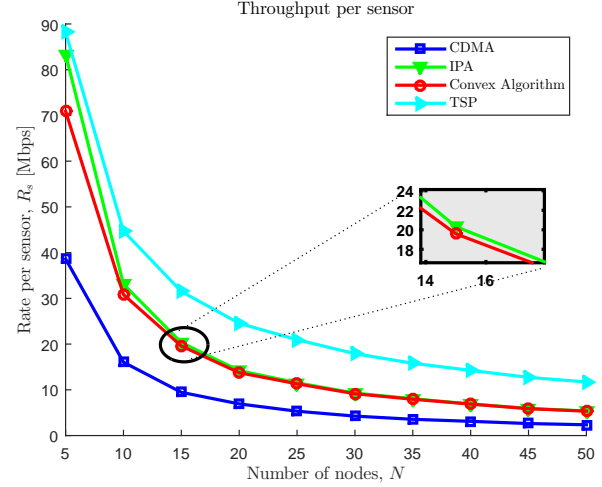


Fig. 12. Bitrate per sensor, R_s , of proposed methods and CDMA approach as a function of N . Notice that the proposed methods yield better performance relative to CDMA.

TABLE IV
MAPPING OF COLORS TO NODES IN FIG. 9 AND FIG. 13.

Message packet from node	Color
1	Green
2	Red
3	Cyan
4	Magenta
5	Yellow
6	Black

B. Random geometric formation with outliers

In this section, we will study the performance of geometric formations of the sensor network with a few sensor nodes far apart from the rest. To create this topology, we construct N nodes distributed according to a mixture of two Gaussian distributions. These distributions are as given below.

$$(x, y) = (\mathcal{N}(0, \sigma^2), \mathcal{N}(0, \sigma^2)) \quad (38)$$

$$(x_o, y_o) = (\mathcal{N}(0, \sigma_o^2), \mathcal{N}(0, \sigma_o^2)) \quad (39)$$

The node location is selected from (38) with probability of 2/3, and from (39) with probability 1/3. We set $\sigma = 5$ and $\sigma_o = 30$; thus for a large N , 1/3 of the nodes will be outliers. A typical topology with 6 nodes is shown in Fig. 13a.

The transmission schedules for interference mitigation, by solving the CA, IPA, and TSP are given in Table V. With these delays introduced, the received packets will not interfere with each other. The received packets at each node are shown in Fig. 13b, Fig. 13c and 13d for the CA, IPA and the TSP algorithms proposed in the paper. Each color in Fig. 13 is mapped to the message from a specific node, as shown in Table IV. For the example network shown in Fig. 13a, the report cycles are approximately given by 473.45 ns, 423.57 ns, and 329.63.55 ns for the CA, TSP, and IPA respectively.

To assess the performance over a large number of nodes N , we performed Monte-Carlo simulations similar to the no-outlier case with 32 distinct random geometric formations constructed from the mixture distribution of (38) and (39) with

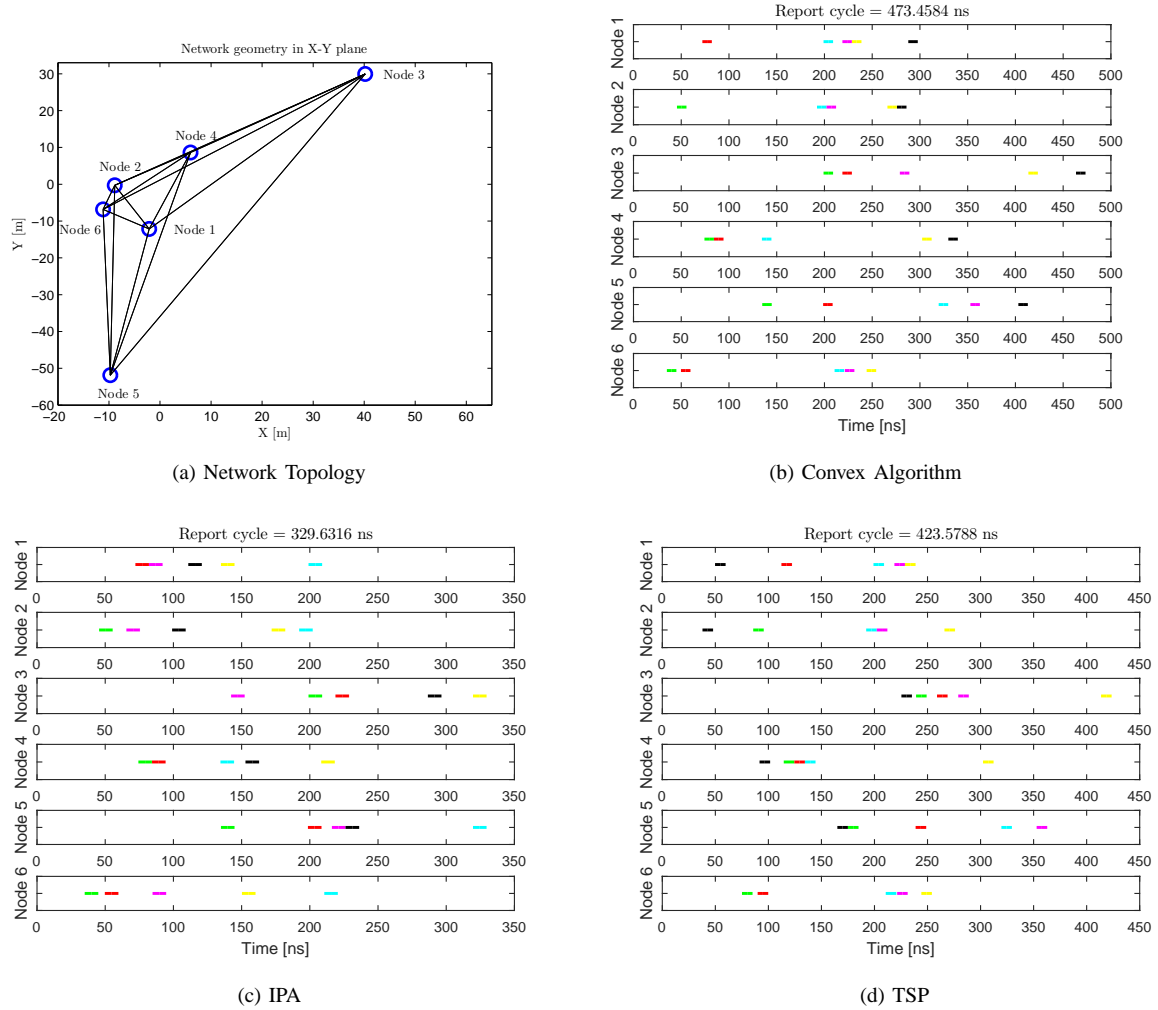


Fig. 13. Network geometry with nodes scattered randomly in a 2-D plane with outliers. Received message packets at each node after introducing the computed delays from Table V for the network topology in Fig. 13a are shown for different algorithms. Notice that the message packets do not interfere. The color of the message packet is mapped to the node as shown in Table IV.

TABLE V
COMPUTED DELAY VALUES FROM PROPOSED METHODS FOR THE
GEOMETRIC FORMATION DEFINED IN FIG. 13A.

Node	Delay Values (Δ_i s) [ns]		
	CA	IPA	TSP
1	0	0	40.16
2	26.60	26.6	66.76
3	0	0	0
4	144.47	7.81	144.47
5	93.94	0	93.94
6	252.85	75.94	14.77

probabilities of 2/3 and 1/3 respectively. The average report cycle is reported in Fig. 14.

Fig. 14 compares the proposed algorithms to the technique of orthogonalization through scheduled transmission discussed in Section II-A. Notice that for a network with 100 nodes, T_R is on average reduced to 1/8 using TSP algorithm. This means that the net communication or update rate can be increased by a factor of 8 on average for the network topology with outliers. Thus, a sensor network with geometric formations having a few outlier nodes can have higher communication rate using

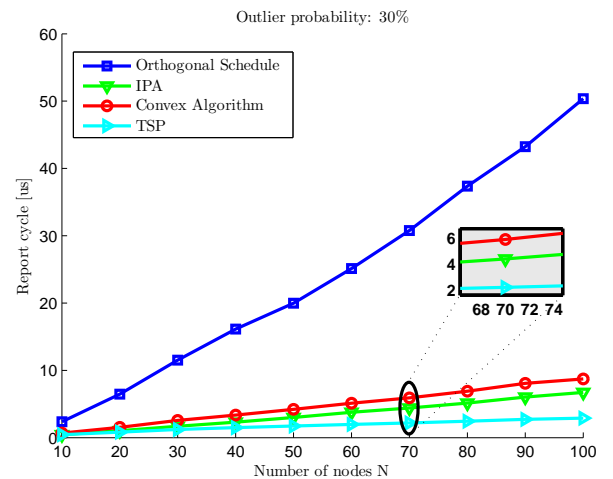


Fig. 14. Performance of proposed methods as a function of N . Notice that as the number of nodes increases, the proposed techniques yield better performance relative to the orthogonal schedule.

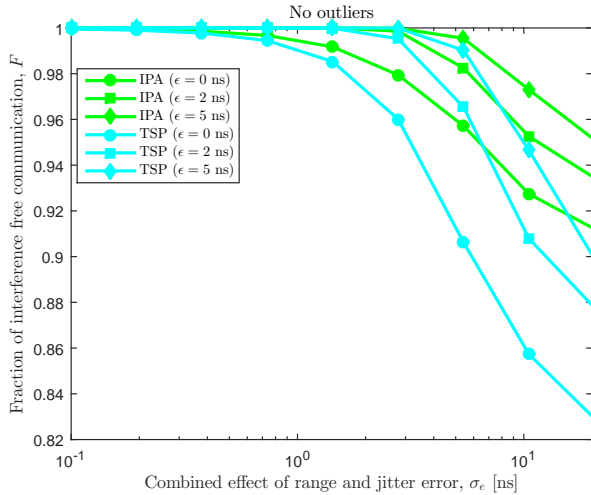


Fig. 15. Interference in the system of 20 Nodes, in presence of imprecise range and clock jitter. By increasing the guard interval, ϵ , by setting higher \mathcal{P} , as in (30), the interference performance can be traded with report cycle. For large σ_e , the collisions are unavoidable despite the guard interval.

the proposed algorithms. The further away these outlier nodes are, the greater the benefits will be, as the algorithms can pack the information packets more efficiently there by optimally utilizing the shared common channel.

C. Performance in the presence of synchronization and range errors

With the configurations for the no-outlier topology as discussed in the Section VI-A, we performed Monte-Carlo simulations to assess the average sensitivity of the algorithms to range and synchronization errors. If there were no synchronization and range errors, then the network would have exchanged $N(N-1)$ packets of width τ , without any interference, using the proposed algorithms. However, due to the errors, packets can interfere and we define the fraction of interference free communication in the network, F , as

$$F = 1 - \frac{\sum_i \mathcal{I}_i}{N(N-1)\tau} \quad (40)$$

where \mathcal{I}_i , denotes the overlapped area of the packets at the receiving node i . The trade-off between the guard interval, ϵ , and report cycle, T_R , is to first pack the transmissions as closely as possible to reduce T_R and then increase the guard interval, ϵ , in the optimization problem, based on the environment to decrease the sensitivity to range and synchronization errors. Fig. 15, shows the average performance of F , using Monte-Carlo simulations for 20 nodes with 100 distinct topologies constructed using the no-outlier case described earlier, with $\tau = 10$ [ns].

VII. CONCLUSION

In this paper, we proposed a methodology for utilizing the range information to arrive at transmission schedules for high density sensor networks. Connected sensor networks need high rates of communication on a shared channel in order to

have high update rates. Therefore, an optimal schedule for accessing the shared common channel needs to be designed for efficient communication. To accomplish this, an optimization problem is formulated using range information for interference mitigation. A solution for the optimization problem is found by CA, TSP and IPA methods. The proposed methods are compared to the traditional time-sharing technique of separating consecutive transmissions by a time interval equal to the duration of maximum path delay in a network. The performances of the algorithms are assessed for different types of networks with varied sizes. Two different geometric formations are considered, one with a random placement of nodes with no outliers and one with outliers. The results are demonstrated in Fig. 10 and Fig. 14. A comparison with traditional CDMA based multiple access is also presented in Fig. 12. The analysis of performance degradation due to non-idealities such as synchronization and range errors is reported in Fig. 15.

The three proposed algorithms performs better than CDMA or orthogonalization by scheduling one node for maximum path delay in the network. As demonstrated in Fig. 14, the performance gains are higher, if the network have few outliers in them. Each of the proposed algorithms has a clear edge over others depending on the type of the network. For example, TSP performs better than IPA and CA for general networks, however, when the network is geometrically larger in relation to the path equivalent message length, \mathcal{L} , then IPA performs better than the TSP and CA as suggested by Fig.11. IPA never performs worse than the convex algorithm. From the simulation results, it appears that the IPA will give same solution as the CA in the worst case scenario, however, the formal proof is not known to the authors.

Table II summarizes the average complexities of the algorithms. For IPA and TSP, the worst case complexities are not known and for CA, the worst case and the average case complexities are same. Thus, IPA and TSP may not be useful in networks where real-time guarantees are needed for the schedule computations. The impact of the synchronization and range errors on the algorithms are studied. As expected, the tighter schedules are more susceptible to the interference due to the imperfect ranging and synchronization. Typical system design involves, first packing the transmissions as closely as possible to have low report cycle using the algorithms discussed and then increase guard interval, ϵ , to decrease the interference as illustrated in Fig. 15.

The performances of the proposed methods are demonstrated in simulations in order to assess the performance gains without platform or network dependencies. The in-house transceiver developed in our lab can yield very precise range information on the order of a few centimeters, as reported in [33]. These transceivers could be mounted on the sensors for joint ranging and communication [33], [34], [41]. The results from the simulations of the proposed schemes indicate that a significant improvement in performance in terms of communication rate can be achieved by using the proposed schemes. With these findings, we intend to further develop the work to implement the schemes on our in-house transceiver hardware and evaluate the performance of in-house transceiver

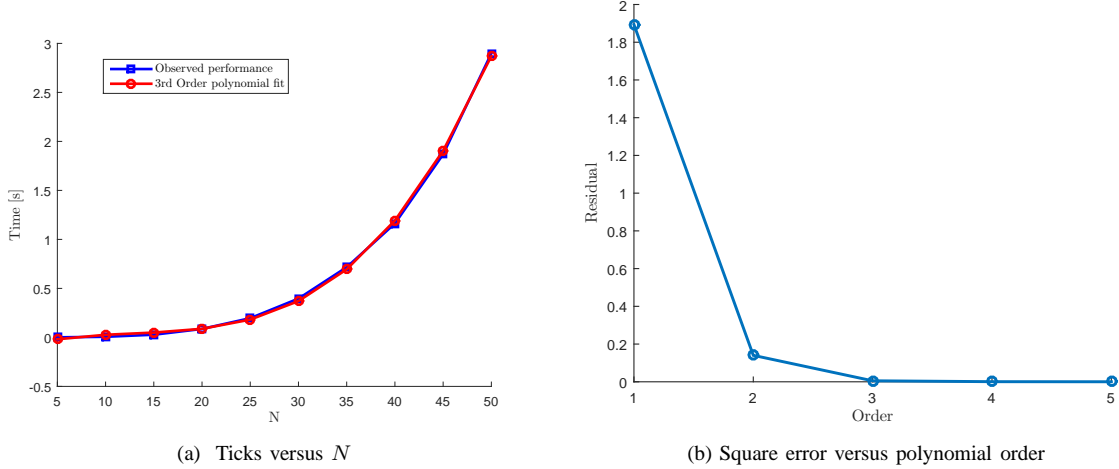


Fig. 16. The 3rd order polynomial fit to the average complexity curve, along with the residual error for the various order polynomial fit.

hardware mounted sensor networks with different sizes with varied geometric formulations.

APPENDIX A

PROOF THAT NODES $i - 1$ AND $i + 1$ DO NOT INTERFERE IN THE CONVEX PROBLEM

In this appendix, we will prove

$$\Delta_{i-1} + \delta_{i,i-1} + \tau \leq \Delta_{i+1} + \delta_{i,i+1}, \quad (41)$$

which says that messages from node $i - 1$ and node $i + 1$ cannot interfere when they are received at node i in the convex problem. To do this, we consider the constraint in the convex optimization problem, for node $i - 1$ and node i , and get

$$\Delta_{i-1} + \delta_{k,i-1} + \tau \leq \Delta_i + \delta_{k,i} \text{ for } k \neq i-1, i \quad (42)$$

and

$$\Delta_i + \delta_{k,i} + \tau \leq \Delta_{i+1} + \delta_{k,i+1} \text{ for } k \neq i, i+1. \quad (43)$$

By setting $k = i + 1$ in (42) and $k = i - 1$ in (43) we get the following two equations

$$\Delta_{i-1} + \delta_{i+1,i-1} + \tau \leq \Delta_i + \delta_{i+1,i} \quad (44)$$

$$\Delta_i + \delta_{i-1,i} + \tau \leq \Delta_{i+1} + \delta_{i-1,i+1}. \quad (45)$$

Adding the left hand sides and the right hand sides of (44) and (45), and using the fact that $\delta_{ik} = \delta_{ki}$ we get

$$\Delta_{i-1} + \delta_{i,i-1} + 2\tau \leq \Delta_{i+1} + \delta_{i,i+1}. \quad (46)$$

This shows that (41) holds and it also shows that the inequality is always strict given that $\tau > 0$.

APPENDIX B

AVERAGE COMPLEXITY OF IPA

In this Appendix, we try to arrive at the average complexity of the IPA algorithm. We first create a random topology, by scattering the nodes in a two-dimensional plane. The

coordinates (x, y) are drawn from a Gaussian distribution as shown below.

$$(x, y) \sim (\mathcal{N}(0, \sigma^2), \mathcal{N}(0, \sigma^2)) \quad (47)$$

We sweep the number of nodes, N , from 5 to 50 and for each N , we calculate the average time elapsed (ticks) for the IPA algorithm for 32 distinct topologies constructed using $\sigma = 5$ [m]. The elapsed times are measured on Intel Core i5 4300U CPU at 1.9 GHz machine with 8 GB RAM running the Windows 8.1 pro operating system. The blue curve in Fig. 16a shows the average time elapsed versus N for IPA. To get the average complexity, we try to fit this graph with polynomials of various orders. We also evaluate the residual square error (square of the L2-norm of the residual vector) for various order polynomial fits, this is shown in Fig. 16b, which indicates that the residual error becomes negligible for an order 3 or more polynomial fit. The red curve in Fig. 16a, shows that the 3rd order polynomial, $1 \times 10^{-4}x^3 - 2.1 \times 10^{-3}x^2 + 3.07 \times 10^{-2}x - 0.12$, gives a perfect fit, indicating that the average complexity of the IPA algorithm is $O(N^3)$.

REFERENCES

- [1] J. Hill, M. Horton, R. Kling, and L. Krishnamurthy, "The platforms enabling wireless sensor networks," *Commun. ACM*, vol. 47, no. 6, pp. 41–46, June 2004.
- [2] I.-K. Rhee, J. Lee, J. Kim, E. Serpedin, and Y.-C. Wu, "Clock Synchronization in Wireless Sensor Networks: An Overview," *Sensors*, vol. 9, no. 1, pp. 56–85, 2009.
- [3] T. Nagayama and B. S. Jr., "Structural health monitoring using smart sensors," Tech. Rep., NSEL Report, Series 001, 2007, [online] Available : <http://hdl.handle.net/2142/3521>.
- [4] M. Anderson and A. Robbins, "Formation flight as a cooperative game," in *Guidance, Navigation, and Control and Co-located Conferences*, pp. 244–250. American Institute of Aeronautics and Astronautics, Aug. 1998.
- [5] F. Mondada, L. Gambardella, D. Floreano, S. Nolfi, J.-L. Deneuborg, and M. Dorigo, "The cooperation of swarm-bots: physical interactions in collective robotics," *Robotics Automation Magazine, IEEE*, vol. 12, no. 2, pp. 21–28, June 2005.
- [6] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, "Towards a swarm of agile micro quadrotors," *Autonomous Robots*, vol. 35, no. 4, pp. 287–300, 2013.

- [7] J. O. Nilsson, D. Zachariah, I. Skog, and P. Händel, "Cooperative localization by dual foot-mounted inertial sensors and inter-agent ranging," *EURASIP Journal on Advances in Signal Processing* 2013, vol. Dec. 2013, pp. 164–174, 2013.
- [8] J. Rantakokko, J. Rydell, P. Strömbäck, P. Händel, J. Callmer, D. Tornqvist, F. Gustafsson, M. Jobs, and M. Gruden, "Accurate and reliable soldier and first responder indoor positioning: multisensor systems and cooperative localization," *Wireless Communications, IEEE*, vol. 18, no. 2, pp. 10–18, Apr. 2011.
- [9] J. Nembrini, N. Reeves, E. Poncet, A. Martinoli, and A. Winfield, "Mascarillons: flying swarm intelligence for architectural research," in *Swarm Intelligence Symposium, 2005. SIS 2005. Proceedings 2005 IEEE*, June 2005, pp. 225–232.
- [10] A. Bharathidasan and V. Anand, "Sensor Networks: An Overview," Tech. Rep., Dept. of Computer Science, University of California at Davis, 2002.
- [11] S. Ganerwal, R. Kumar, and M. B. Srivastava, "Timing-sync Protocol for Sensor Networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, New York, NY, USA, 2003, SenSys '03, pp. 138–149, ACM.
- [12] J. Elson, L. Girod, and D. Estrin, "Fine-grained Network Time Synchronization Using Reference Broadcasts," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 147–163, Dec. 2002.
- [13] S. Dwivedi, A. De Angelis, D. Zachariah, and P. Händel, "Joint Ranging and Clock Parameter Estimation by Wireless Round Trip Time Measurements," *Selected Areas in Communications, IEEE Journal on*, vol. PP, no. 99, pp. 1–1, 2015.
- [14] R. Rajan and A. van der Veen, "Joint Ranging and Synchronization for an Anchorless Network of Mobile Nodes," *Signal Processing, IEEE Transactions on*, vol. 63, no. 8, pp. 1925–1940, April 2015.
- [15] T. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall communications engineering and emerging technologies series. Dorling Kindersley, 2009.
- [16] T. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. Wong, J. Schulz, M. Samimi, and F. Gutierrez, "Millimeter Wave Mobile Communications for 5G Cellular: It Will Work!," *Access, IEEE*, vol. 1, pp. 335–349, 2013.
- [17] P. Agyapong, M. Iwamura, D. Staehle, W. Kiess, and A. Benjebbour, "Design considerations for a 5G network architecture," *Communications Magazine, IEEE*, vol. 52, no. 11, pp. 65–75, Nov. 2014.
- [18] B. Panzner, W. Zirwas, S. Dierks, M. Lauridsen, P. Mogensen, K. Pajukoski, and D. Miao, "Deployment and implementation strategies for massive MIMO in 5G," in *Globecom Workshops (GC Wkshps), 2014*, Dec. 2014, pp. 346–351.
- [19] D. Bertsimas and J. Tsitsiklis, *Introduction to Linear Optimization*, Athena Scientific, 1st edition, 1997.
- [20] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, New York, NY, USA, 2004.
- [21] E. Lawler, *Combinatorial Optimization: Networks and Matroids*, Saunders College Publishing, Fort Worth, 1976.
- [22] L. Lovász, "Combinatorial Problems and Exercises. 1979," *Akadémiai Kiadó, Budapest*.
- [23] E. Lawler, *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, Wiley, New York, 1985.
- [24] D. S. Johnson, "The NP-completeness column: An ongoing guide," *Journal of Algorithms*, vol. 3, no. 2, pp. 182 – 195, 1982.
- [25] G. Laporte, "The traveling salesman problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, no. 2, pp. 231 – 247, 1992.
- [26] K. Helsgaun, "An effective implementation of the Lin–Kernighan traveling salesman heuristic," *European Journal of Operational Research*, vol. 126, no. 1, pp. 106–130, 2000.
- [27] C. R. Reeves, Ed., *Modern Heuristic Techniques for Combinatorial Problems*, John Wiley & Sons, Inc., New York, NY, USA, 1993.
- [28] D. J. Rosenkrantz, R. E. Stearns, and I. Philip M. Lewis, "An Analysis of Several Heuristics for the Traveling Salesman Problem," *SIAM Journal on Computing*, vol. 6, no. 3, pp. 563–581, 1977.
- [29] J. Perttunen, "On the Significance of the Initial Solution in Travelling Salesman Heuristics," *The Journal of the Operational Research Society*, vol. 45, no. 10, pp. 1131–1140, 1994.
- [30] S. Dwivedi, D. Zachariah, A. De Angelis, and P. Händel, "Cooperative Decentralized Localization Using Scheduled Wireless Transmissions," *Communications Letters, IEEE*, vol. 17, no. 6, pp. 1240–1243, June 2013.
- [31] S. Gezici, Z. Tian, G. Giannakis, H. Kobayashi, A. Molisch, H. Poor, and Z. Sahinoglu, "Localization via Ultra-Wideband Radios: A look at positioning aspects for future sensor networks," *Signal Processing Magazine, IEEE*, vol. 22, no. 4, pp. 70–84, July 2005.
- [32] D. Dardari, C.-C. Chong, and M. Win, "Threshold-Based Time-of-Arrival Estimators in UWB Dense Multipath Channels," *Communications, IEEE Transactions on*, vol. 56, no. 8, pp. 1366–1378, Aug. 2008.
- [33] A. De Angelis, S. Dwivedi, and P. Händel, "Characterization of a Flexible UWB Sensor for Indoor Localization," *Instrumentation and Measurement, IEEE Transactions on*, vol. 62, no. 5, pp. 905–913, 2013.
- [34] V. Yajnanarayana, S. Dwivedi, A. De Angelis, and P. Händel, "Spectral efficient IR-UWB communication design for low complexity transceivers," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, pp. 158, 2014.
- [35] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 147–163, Dec. 2002.
- [36] S. PalChaudhuri, A. K. Saha, and D. B. Johnson, "Adaptive clock synchronization in sensor networks," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, New York, NY, USA, 2004, IPSN '04, pp. 340–348, ACM.
- [37] P. Bahl and V. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," in *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 2000, vol. 2, pp. 775–784.
- [38] R. W. Jeffrey Hightower, Gaetano Borriello, "Spoton: An indoor 3d location sensing technology based on rf signal strength," Tech. Rep., UW CSE Technical Report, 2000.
- [39] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: a quantitative comparison," *Computer Networks*, vol. 43, no. 4, pp. 499 – 518, 2003, Wireless Sensor Networks.
- [40] W. Lovelace and J. Townsend, "The effects of timing jitter and tracking on the performance of impulse radio," *Selected Areas in Communications, IEEE Journal on*, vol. 20, no. 9, pp. 1646–1651, Dec. 2002.
- [41] V. Yajnanarayana, S. Dwivedi, A. De Angelis, and P. Händel, "Design of impulse radio UWB transmitter for short range communications using PPM signals," in *Electronics, Computing and Communication Technologies (CONECT), 2013 IEEE International Conference on*, 2013, pp. 1–4.